IN THE CLAIMS:

The following listing of claims replaces all previous claim listings and versions.

1. – 10. (Canceled)

11. (Currently Amended) A method in a LSI design and development process for evaluating an architecture design for an algorithm design by performing a performance evaluation of at least one bus at a high-level stage of said design and development process, said method comprising:

structuring source code describing the algorithm design in a general purpose high-level programming language by isolating elements of said source code representing hardware units and software units;

creating an evaluation function for counting data traffic that occurs on said at least one bus, the bus being a part of the source code realizing the data traffic between said elements representing hardware units and software units;

sequentially reading in the source code line by line while effecting syntax analysis;

determining whether the source code is to be modified based on whether a line of source code represents writing data to variables that are defined in advance and are loaded onto the bus to be evaluated;

modifying at least one element of said source code elements based on a result of an implementation of said evaluation function; and

performing said performance evaluation by simulating said modified source code elements and counting said data traffic on the bus~~; and~~

~~in response to said performance evaluation, modifying the configuration of said bus~~.

12. (Previously Presented) The method according to claim 11, further comprising:

restructuring the source code based on the evaluated data traffic; and

performing said performance evaluation again by simulating said restructured source code again.

13. (Previously Presented) The method according to claim 11, wherein a bus traffic is calculated from the evaluated data traffic with respect to the processing rate of the bus.

14. (Previously Presented) The method according to claim 11, further comprising:

feeding back a result of the performance evaluation of the bus to the step of structuring the source code to improve the architecture design at a high-level design stage by isolating in said source code new elements representing hardware units and new elements representing software units.

15. (Previously Presented) The method according to claim 14, wherein in response to the bus traffic, isolation of the source code into elements representing hardware units and elements representing software units is optimized.

16. (Currently Amended) The method according to claim 11, further comprising:

~~after creating the evaluation function, sequentially reading in the source code line by line while effecting syntax analysis;~~

~~determining whether the source code is to be modified based on whether a line of source code represents writing data to variables that are defined in advance and are loaded onto the bus to be evaluated;~~

upon determining that the source code is to be modified, modifying the source code by embedding the evaluation function one of immediately before or immediately after the line of source code in which the variable is written;

repeating the forgoing steps until the source code is completely read in up to a last line of source code;

structuring the source code into elements representing at least one of the hardware units and the software units for use in the architecture design by compiling said source code;

calculating the data transfer rate on the bus by executing the compiled source code elements in a simulation program;

calculating bus traffic with regard to a given processing rate of the bus; and

performing evaluation of the performance of the bus in response to the bus traffic.

17. (Previously Presented) The method according to claim 16, wherein the variables loaded onto the bus consist of n bits while the bus consists of m bit lines, where n and m are both integers, and n is a multiple of m, and the bus traffic for the processing rate is produced such that the number of times in effecting data transfer on the bus is multiplied by n/m and is then divided by the processing rate.

18. (Previously Presented) The method according to any one of the preceding claims, wherein the general purpose high-level programming language is one of C language and C++ language.

19. (Previously Presented) The method according to claim 11,wherein the evaluation function increments a counting value if a pre-defined variable is loaded onto the bus.

20. (Previously Presented) A method in a LSI design and development process for evaluating and facilitating an architectural design for an algorithm design by performing a performance evaluation of at least one bus at said architecture design being a high-level stage of said design and development process, said method comprises the steps of:

structuring a source code describing the algorithm design in a general purpose high-level programming language by isolating said source code in hardware and software elements interconnected by said bus having a given bus configuration;

creating an evaluation function for evaluating data transfer that occurs on said bus, wherein said evaluation function counts and represents a number of times the source code effecting said data transfer on said bus;

sequentially reading in said source code;

determining whether a line of source code represents writing data onto the bus to be evaluated;

upon the determination, modifying the source code by embedding the evaluation function just before or after the line of source code in which the variable is written;

repeating the forgoing steps until the source code is completely read in and modified up to a last line of source code;

simulating said modified source code elements at the architecture design level and evaluating said data transfer on the bus, comprising the steps of:

compiling said structured source code elements;

executing the compiled source code elements on a simulation platform;

calculating the bus traffic based on the number of times in effecting of data transfers according to the evaluation function and a given processing rate that is already known;

performing the evaluation of the performance of the bus in response to the bus traffic being produced; and

wherein said method is carried out again in response to the performance evaluation if necessary by starting with changing said bus configuration and restructuring the source code.

21. (Previously Presented) The method according to claim 20, further comprising:

feeding back the result of the performance evaluation of the bus to the step of structuring the source code to improve the architecture design at a high-level design stage by re-isolating said source code in hardware and software elements.

22. (Previously Presented) The method according to claim 20, wherein the bus configuration comprises the number of bit lines of the bus and, in response to the bus traffic, isolation of the source code in hardware and software elements is optimized by changing the number of bit lines of the bus.

23. (Previously Presented) The method according to claim 20, further comprising:

after creating the evaluation function, sequentially reading in the source code line by line while effecting syntax analysis.

24. (Previously Presented) The method according to claim 20, wherein the variables loaded onto the bus consist of n bits while the bus consists of m bit lines, where n, m are both integral numbers, and n >= m, and the bus traffic for the processing rate is produced such that the number of times in effecting data transfer on the bus is multiplied by n/m and is then divided by the processing rate.

25. (Previously Presented) The method according to claim 20, wherein the general purpose high-level programming language is C language or C++ language.

26. (Previously Presented) The method according to claim 20, wherein the evaluation function is to increment a counting value if a pre-defined variable is loaded onto the bus.